Serial Peripheral Interface (SPI) Host Controller Data Sheet

Proven System Block (PSB) for QuickLogic Customer Specific Standard Products (CSSPs)

Features

- Supports Master configuration (Multi-Master configuration is not supported).
- Capable of being connected to eight SPI slaves with individual slave select lines.
- Supports Normal and Bidirectional modes of operation.
- Interrupt generation capability (transmitter and receiver operations are accomplished through interrupts to the processor from either the Transmit or Receive FIFO.
- Serial clock with programmable phase and polarity.
- Supports speeds of up to one-half of the system clock.
- Four programmable transfer formats supported (controlled by CPOL and CPHA).
- Buffered operation with separate transmit and receive FIFO – 1024-byte deep for post preamble data transfer.
- Maximum bit rate of 20 Mbits/sec. with a system clock of 40 MHz.
- 1.8/2.5/3.3 V I/O voltages are supported.
- Supports up to 4 bytes of preamble (for transferring the address and the instructions).
- Arbitrary choice of message size, contents, and purpose.

Overview

The QuickLogic SPI Host controller can communicate with up to eight SPI slave devices. To achieve a high throughput of 20 Mbits/sec., the SPI controller utilizes a 32-bit processor interface and two 8-Kbit FIFOs (Tx and Rx). The microprocessor interface can push/pop data 32 bits at a time, while the SPI transmits/receives 8 bits at a time.

Transmitter and receiver operations are accomplished through interrupts to the processor from either the Transmit or Receive FIFO. The Write Interrupt is triggered when the Transmit FIFO is empty and the Read Interrupt is triggered when the received bytes count equals the preset value.

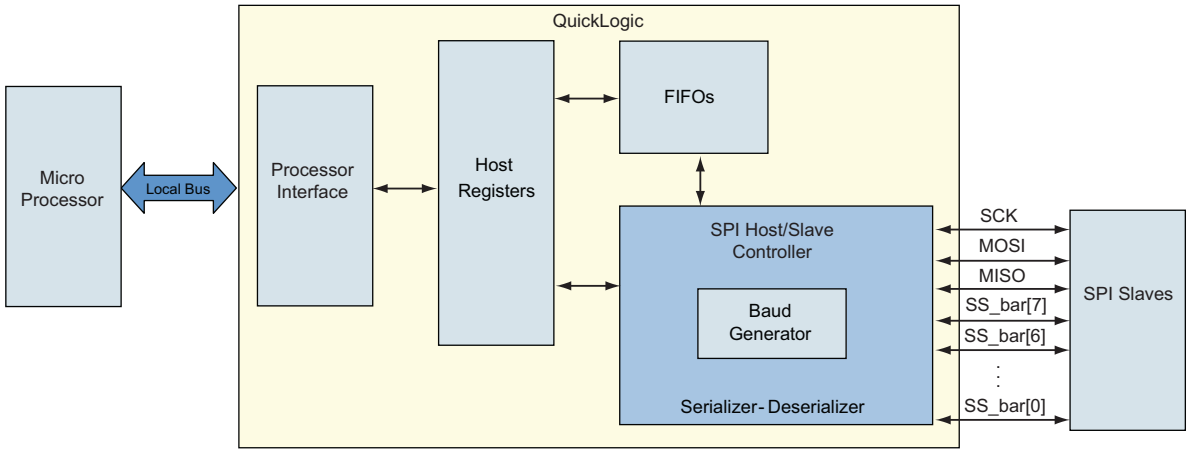
SPI Host Controller Architecture

The SPI Host Controller is a full-duplex synchronous serial input/output data connection that is standard on many embedded processors. It is traditionally used to interact with slow peripheral devices such as EEPROMs, real-time clocks and analog-to-digital converters (ADCs). However with the increasing speed, SPI also evolved to become a standard interface to wireless LAN 802.11a/b/g as well as mobile TV chipsets.

QuickLogic SPI Host controller is designed to interface with different CPU interfaces and up to eight SPI slave devices. The internal high speed serializer/deserializer and FIFOs ensure high data rate of 20 Mbit/sec. to meet WiFi and mobile television requirements.

Figure 1 shows a top-level block diagram of the QuickLogic SPI Host Controller. The microprocessor interface is customizable to any microprocessor.

Figure 1: SPI Host Controller Architecture Block Diagram



Interface List and Description

The SPI module has a total of 11 external pins for the slave interface and 43 for the processor interface. The processor interface is customizable for different processor interfaces. **Table 1** shows an example pin description for the PXA270 processor which includes the name, type, description of the ports, and the inputs/outputs that can connect off chip.

Table 1: SPI PXA270 Pin Description

Pin	Type	Description
PXAxxx VLIO Interface		
CPU_DATA[31:0]	I/O	VLIO data bus.
CPU_ADDR[4:2]	I	VLIO address bus. Signals the address requested for memory accesses.
CPU_RDnWR	I	VLIO Interface Read /not Write indicator.
CPU_RDY	I	VLIO active high Ready signal.
CPU_nBE[3:0]	I	VLIO active low Byte Enable Bus.

Table 1: SPI PXA270 Pin Description *(Continued)*

Pin	Type	Description
CPU_CS	I	VLIO active low Chip Select.
CPU_INT	O	VLIO interrupt.
SPI Interface		
SCK	I/O	Serial Clock (output from the Master).
SS_bar0	I/O	Slave Select 0 (active low, output from the Master).
SS_bar1	I/O	Slave Select 1 (active low, output from the Master).
SS_bar2	I/O	Slave Select 2 (active low, output from the Master).
SS_bar3	I/O	Slave Select 3 (active low, output from the Master).
SS_bar4	I/O	Slave Select 4 (active low, output from the Master).
SS_bar5	I/O	Slave Select 5 (active low, output from the Master).
SS_bar6	I/O	Slave Select 6 (active low, output from the Master).
SS_bar7	I/O	Slave Select 7 (active low, output from the Master).
MOSI	I/O	Master Output, Slave Input (output from the Master).
MISO	I/O	Master Input, Slave Output (output from the Slave).

Register Sets

Table 2 shows the memory map of the SPI register set. There are five registers that control the SPI operation. The address listed for each register is the offset from the base address of the chip select of the processor.

Table 2: SPI Register Set Memory Map

Address	Description	Access
0x00	SPI Configuration Register	Read/Write[a]
0x04	Reserved	
0x08	Preamble Register	Write
0x0C	SPI Transfer Register	Write
0x10	SPI Interrupt Register	Read
0x14	Slave Select Register	Write
0x18	Write FIFO Port	Write
0x1C	Read FIFO Port	Read

a. Some bits are read-only.

SPI Configuration Register (offset 0x00)

- Select SCK phase and polarity
- Master and Slave modes
- Interrupt flag enable
- Baud rate setting
- Normal/Bidirectional mode of operation

Preamble Register (offset 0x08)

- This register is optional and can contain a combination of address bytes and opcode

SPI Transfer Register (offset 0x0C)

- Start/Stop control
- Number of bytes to be transmitted/received
- Read/Write control
- Byte enable for the Preamble Register

SPI Interrupt Register (offset 0x10)

- Read Interrupt
- Write Interrupt
- Processor Interrupt

Slave Select Register (offset 0x14)

- Eight external slave select bits

Table 3 shows the SPI register set in more detail.

Table 3: Register Set

Byte 3	Byte 2	Byte 1	Byte 0	Offset
SPISR	SPIBR	SPICR2	SPICR1	0x00
UNUSED (Reserved)				0x04
PREAMBLE				0x08
	BE[3], BE[2], BE[1], BE[0]	# of data_bytes[15:3]	R, W, S/Sp	0x0C
			IP, IW, IR	0x10
			S7, S6, S5, S4, S3, S2, S1, S0	0x14
WRITE FIFO PORT				0x18
READ FIFO PORT				0x1C

Register Descriptions

This section provides a detailed description of all the registers in the SPI Host Controller and the corresponding bits.

Register: SPICR1 (Offset 0x00, Byte 0)

The SPI control register 1(SPICR1) should be written to only when a transaction is not in progress. This register configures the SPI Host Controller. It controls the mode of operation (Master/Slave), the phase and polarity of the clock, and the transfer type (LSB or MSB first).

Table 4: SPI Control Register 1 (SPICR1)

Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
SPIE	SPE	Rsv	MSTR	CPOL	CPHA	Rsv	LSBFE

Bit 1 and Bit 5 are not implemented (reads or writes to these bits are not implemented). They are reserved for future use. After system reset the following register bits are set to the default values.

Table 5: SPICR1 Bits

0x00[7:0] Bits	Type	Description	Default Value
Bit[0]	R/W	**LSBFE:** This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7. 1 = Data is transferred LSB first. 0 = Data is transferred MSB first.	0
Bit[1]	-	N/A	0
Bit[2]	R/W	**CPHA:** This bit is used to shift the SCK serial clock. 1 = The first SCK edge is issued at the beginning of the 8-cycle transfer operation. 0 = The first SCK edge is issued one-half cycle into the 8-cycle transfer operation.	1
Bit[3]	R/W	**CPOL:** SPI clock polarity bit. This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. 1 = Active-low clocks selected; SCK idles high. 0 = Active-high clocks selected; SCK idles low.	0
Bit[4]	R/W	**MSTR:** SPI Master/Slave mode select bit. 1 = Master mode. 0 = Slave mode.	0

Table 5: SPICR1 Bits *(Continued)*

0x00[7:0] Bits	Type	Description	Default Value
Bit[5]	-	N/A	0
Bit[6]	R/W	**SPE:** SPI system enable bit. This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. 1 = SPI port pins are dedicated to SPI functions. 0 = SPI disabled (lower power consumptions).	0
Bit[7]	R/W	**SPIE:** SPI interrupt enable bit. This bit enables SPI interrupts each time the SPIF or MODF status flag is set. 1 = SPI interrupts enabled. 0 = SPI interrupts disabled.	0

Register: SPICR2 (Offset 0x00, Byte 1)

The SPI control register 2(SPICR2) should be written to only when a transaction is not in progress. This register controls the mode of operation (Normal/Bidirectional) and enables the bidirectional pin configuration.

Table 6: SPI Control Register 2 (SPICR2)

Bit[15]	Bit[14]	Bit[13]	Bit[12]	Bit[11]	Bit[10]	Bit[9]	Bit[8]
Rsv	Rsv	Rsv	Rsv	Rsv	BIDIROE	Rsv	SPC0

Bit[7:3] and Bit[1:0] are not implemented (reads or writes to these bits are not implemented). They are reserved for future use. After system reset the following register bits are set to the default values.

Table 7: SPICR2 Bits

0x00[15:8] Bits	Type	Description	Default Value
Bit[8]	R/W	**SPC0:** Serial pin control bit 0. With the MSTR control bit, this bit enables bidirectional pin configurations as shown in **Table 8**.	0
Bit[9]	-	N/A	0
Bit[10]	R/W	**BIDIROE:** Output enable in the bidirectional mode. This bit along with the MSTR bit of SPCR1 is used to enable the output buffer when the SPI is configured in the bidirectional mode. 1 = Output buffer enabled. 0 = Output buffer disabled.	0
Bit[11]	-	N/A	0
Bit[12]	-	N/A	0
Bit[13]	-	N/A	0
Bit[14]	-	N/A	0
Bit[15]	-	N/A	0

Table 8: Bidirectional Pin Configurations

Ports	Description	Master	
		Normal	Bidir.
SCK	Synchronization clock to the Slave	Out	Out
MOSI	Master Out, Slave In	Out	INOUT
MISO	Master In, Slave Out	In	Not Used
SS_bar	Slave Select signal	Out	Out

Register: SPIBR (Offset 0x00, Byte 2)

This is a Read/Write register and controls the baud rate of the SPI system.

Table 9: SPI Baud Rate Register (SPIBR)

Bit[23]	Bit[22]	Bit[21]	Bit[20]	Bit[19]	Bit[18]	Bit[17]	Bit[16]
Rsv	SPPR2	SPPR1	SPPR0	Rsv	SPR2	SPR1	SPR0

Bit 7 and bit 3 are not implemented (reads or writes to these bits are not implemented). They are reserved for future use. Before using the SPI Host Controller, these register bits must be set to the appropriate values.

Table 10: SPIBR Bits

0x00[23:16] Bits	Type	Description	Default Value
Bit[16]	R/W	**SPR0:** SPI Baud Pre-selection bit 0	0
Bit[17]	R/W	**SPR1:** SPI Baud Pre-selection bit 1	0
Bit[18]	R/W	**SPR2:** SPI Baud Pre-selection bit 2	0
Bit[19]	-	N/A	0
Bit[20]	R/W	**SPPR0:** SPI Baud selection bit 0	0
Bit[21]	R/W	**SPPR1:** SPI Baud selection bit 1	0
Bit[22]	R/W	**SPPR2:** SPI Baud selection bit 2	0
Bit[23]	-	N/A	0

The Baud Rate divisor can be calculated using the following formula:

$$\text{Baud Rate Divisor} = (\text{SPPR} + 1) \cdot 2^{(\text{SPR} + 1)}$$

$$\text{Baud Rate} = \text{Bus clock} / \text{Baud Rate Divisor}$$

Register: SPISR (Offset 0x00, Byte 3)

This read only register contains the transmit empty flag indicating there is room in the transmit data buffer.

Table 11: SPI Status Register (SPISR)

Bit[31]	Bit[30]	Bit[29]	Bit[28]	Bit[27]	Bit[26]	Bit[25]	Bit[24]
Rsv	Rsv	SPTEF	Rsv	Rsv	Rsv	Rsv	Rsv

Table 12: SPISR Bits

0x00[31:24] Bits	Type	Description	Default Value
Bit[24]	-	N/A	0
Bit[25]	-	N/A	0
Bit[26]	-	N/A	0
Bit[27]	-	N/A	0
Bit[28]	-	N/A	0
Bit[29]	R/W	**SPTEF:** SPI transmit empty flag	0
Bit[30]	-	N/A	0
Bit[31]	-	N/A	0

Offset 0x04 Unused (Reserved)

This register is not implemented and is reserved for future use.

Table 13: Unused Register (0x04)

Byte 3	Byte 2	Byte 1	Byte 0
UNUSED (Reserved)			

Preamble Register (Offset 0x08)

This 4-byte optional Preamble register can be a combination of the opcode and the address bytes; depending on the application and the SPI slave protocol, other formats can be used as well.

Table 14: Preamble Register

Preamble Byte 3								Preamble Byte 2								Preamble Byte 1								Preamble Byte 0							

SPI Transfer Register (Offset 0x0C)

This register contains the bit to start/stop a transaction and indicates whether it is a read or write transaction. It also consists of the preamble register byte enables and the number of data bytes to be transmitted or received.

Table 15: SPI Transfer Register (SPITR)

	SPI Transfer Register								
	BE[4]	BE[3]	BE[2]	BE[1]	DATA BYTES [15:3]		R	W	S/Sp

Table 16: SPITR Bits

0x00[31:0] Bits	Description	Default Value
Bit[0]	**S/Sp:** Start/Stop a transaction.	0
Bit[1]	**W:** Performs a write transaction	0
Bit[2]	**R:** Performs a read transaction	0
Bit[15:3]	**data_bytes:** Number of data bytes to be transmitted/received	0
Bit[16]	**BE[1]:** First byte enable of the preamble register	0
Bit[17]	**BE [2]:** Second byte enable of the preamble register	0
Bit[18]	**BE [3]:** Third byte enable of the preamble register	0
Bit[19]	**BE[4]:** Fourth byte enable of the preamble register	0
Bit[31:20]	N/A	0

SPI Interrupt Register (Offset 0X10)

This SPI Interrupt Register (SPIIR) contains Interrupt Flags, indicating the completion of transactions. The three types of interrupt flags include:

- Read Interrupt – indicates a read transaction has been completed
- Write Interrupt – indicates a write transaction has been completed
- Preamble Interrupt –indicates all preamble bytes have been sent

Table 17: SPI Interrupt Register

Interrupt Register			
Rsv	IP	IW	IR

Table 18: SPIIR Bits

0x00[31:0] Bits	Type	Description	Default Value
Bit[0]	RO	**IR:** Interrupt flag that indicates read is done	0
Bit[1]	RO	**IW:** Interrupt flag that indicates write is done	0
Bit[2]	RO	**IP:** Interrupt flag the indicates preamble is done	0
Bit[31:3]	-	N/A	0

Slave Select Register (Offset 0x14)

The SPI slave select register can be used to communicate with up to eight slave devices by setting the corresponding bit. This register configures which slave is selected during a transaction.

Table 19: SPI Slave Select Register

Slave Select Register								
	S7	S6	S5	S4	S3	S2	S1	S0

Table 20: SPI Slave Select Register Bits

0x00[31:0] Bits	Description	Default Value
Bit[0]	**S0:** Slave 1 select bit	0
Bit[1]	**S1:** Slave 2 select bit	0
Bit[2]	**S2:** Slave 3 select bit	0
Bit[3]	**S3:** Slave 4 select bit	0
Bit[4]	**S4:** Slave 5 select bit	0
Bit[5]	**S5:** Slave 6 select bit	0
Bit[6]	**S6:** Slave 7 select bit	0
Bit[7]	**S7:** Slave 8 select bit	0
Bit[31:8]	N/A	0

Functional Description

Figure 2 illustrates the data transmission mechanism between the Master and Slave.

Figure 2: Master/Slave Block Diagram

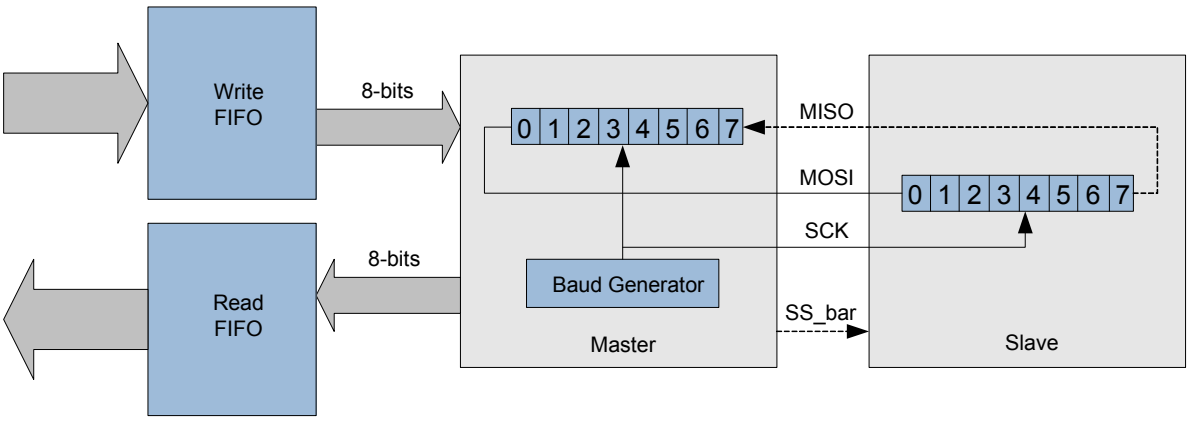


Table 21 shows the various modes of configurations for the SPI Host Controller.

Table 21: SPI Pin Configurations

Ports	Description	Master	
		Normal	Bidir.
SCK	Synchronization clock to the Slave	Out	Out
MOSI	Master Out, Slave In	Out	INOUT
MISO	Master In, Slave Out	In	Not Used
SS_bar	Slave Select signal	Out	Out

Phase and Polarity

The clock polarity control bit CPOL – SPICR1 (offset 0x00, Byte0, bit[3]) and the clock phase control bit CPHA – SPICR1 (offset 0x00, byte0, bit2) in the SPI Configuration Register selects one of the four possible clock modes listed in **Table 22**. The combination of phase and polarity is referred to as a mode.

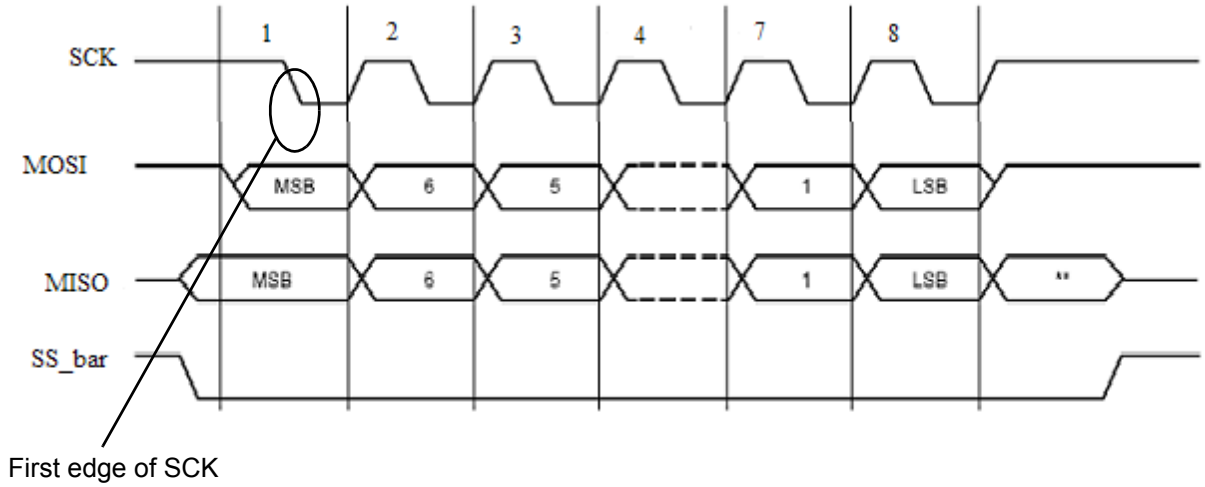
Table 22: SPI Modes[a]

Mode	CPOL	CPHA	Description
1	0	0	SCK idles low, data latched on the clock's rising edge and shifted on the falling edge.
2	0	1	SCK idles low, data latched on the clock's falling edge and shifted on the rising edge.
3	1	0	SCK idles high, data latched on the clock's falling edge and shifted on the falling edge.
4	1	1	SCK idles high, data latched on the clock's rising edge and shifted on the rising edge.

a. For a description of the CPOL and CPHA bits, see **Table 5**.

Figure 3 shows the waveform of a SPI transfer when CPOL = 1 and CPHA = 0. SCK idles high and data is latched on the first edge of SCK.

Figure 3: Data Transfer with CPHA = 0 and CPOL =1



Configuring SPI ports in Normal and Bidirectional Modes

Refer to **Table 21** for more information about modes.

Normal Mode:

- Master configuration:
  - Set MSTR bit (Offset Address 0x00, Byte 0, bit [4]) to "1".
  - Set SPC0 bit (Offset Address 0x00, Byte 1, bit [0]) to "0".
  - Set BIDIROE bit (Offset Address 0x00, Byte 1, bit [3]) to "0".

Bidirectional Mode of Operation:

- Master configuration:
  - Set MSTR bit (Offset Address 0x00, Byte 0, bit [4]) to "1".
  - Set SPC0 bit (Offset Address 0x00, Byte 1, bit [0]) to "1".
  - Set BIDIROE bit (Offset Address 0x00, Byte 1, bit [3]) to "1".

SPI Host Operation

This section describes the configuration and operation of the SPI Host Controller and details how the reads and writes are performed with the external slave device.

Master Read Operation

To perform a master read operation:

1. Set up the SPI controller by writing to the SPI configuration register (offset 0x00).

   NOTE: Steps a through g can be performed with a single write to address offset 0x00.

   a. Enable the SPI by setting the **SPE** bit (byte 0, bit [6]) to "1".

   b. Configure the SPI as a Master by setting the **MSTR** bit (B=byte 0, bit [4]) to "1".

c. Configure the shift-out direction by setting the **LSBFE** bit (byte 0, bit [0]):

- If data is to be transferred least significant bit first, set **LSBFE** to '1'
- If data is to be transferred most significant bit first, set **LSBFE** to '0'

d. Program the baud rate bits in **SPIBR** (Offset Address 0x00, Byte 2). Refer to **Register Descriptions** on page 5.

e. Set the SPI clock phase (**CPHA**, byte 0, bit [2]) and polarity bits (**CPOL**, byte 0, bit [3]) to the desired setting. Refer to the **Phase and Polarity** on page 11.

f. Enable the SPI interrupts by setting the **SPIE** bit (byte 0, bit [7]) to "1".

g. To enable the bidirectional mode of operation, set the **SPC0** bit (byte 1, bit [0]) to "1" and the **BIDIROE** bit (byte 1, bit [3]) to a "1". For normal mode, set both the bits to "0".

2. If applicable, write to the preamble register (offset 0x08) with the desired opcode/address.

3. Select an external slave by asserting the corresponding bit in the Slave Select Register (offset 0x14). For example, to select the slave device connected to the SS_bar [7], set the **S7** bit (byte 0, bit [8]) to "1".

4. Start data transfer by writing to the SPI Transfer Register (offset 0x0C).

NOTE: Steps a through d can be performed with a single write to address offset 0x0C.

a. For a read operation, set the **W** bit (byte 0, bit [1]) to "0" and the **R** bit (byte 0, bit [2]) to "1".

b. Set the preamble register byte enable bits by setting the corresponding **BE**[3:0] bits to "1". For example, if the two most significant bytes of the preamble register are to be transmitted, set **BE**[3:2] (byte 2, bit [3:2]) to "11". If no preamble bytes are required, set **BE**[3:0] to "0000".

c. Configure the number of bytes to be received by setting **data_bytes**[15:3] to the appropriate value. For example, to read 10 bytes, the value of **data_bytes**[15:3] should be "0000000001010".

d. To start the read process, set the **S/Sp** bit (byte 0, bit [0]) to "1".

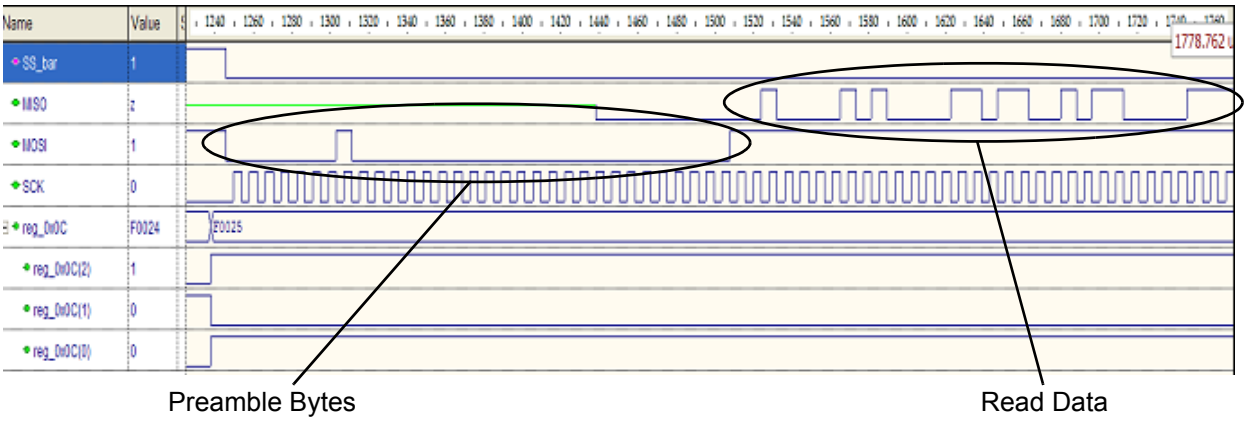
5. After the read transaction is completed (i.e., the desired number of data bytes have been received – 10 bytes for the above example) and put in the receive FIFO, the SPI controller generates an interrupt signal to the processor, **IR** (offset 0x10, byte 0, bit 0).

NOTE: This interrupt signal automatically clears the S/Sp bit (byte 0, bit [0]) to "0".

a. To acknowledge the interrupt, the processor must read the SPI interrupt register (offset 0x100) to clear the read interrupt flag.

b. The processor must then transfer all data from the receive FIFO by reading **data_bytes**[15:3] number of bytes from the **Read_FIFO_Port** (offset 0x1C).

Figure 4: Master Read Operation



Master Write Operation

To perform a Master Write Operation:

1. Set up the SPI controller by writing to the SPI configuration register (offset 0x00).

   NOTE: Steps a through g can be performed with a single write to address offset 0x00.

   a. Enable the SPI by setting the **SPE** bit (byte 0, bit [6]) to "1".

   b. Configure the SPI as a Master by setting the **MSTR** bit (byte 0, bit [4]) to "1".

   c. Configure the shift-out direction by setting the **LSBFE** bit (byte 0, bit [0]).

   - If data is to be transferred least significant bit first, set **LSBFE** to '1'
   - If data is to be transferred most significant bit first, set **LSBFE** to '0'

   d. Program the baud rate bits in **SPIBR** (Offset Address 0x00, Byte 2). Refer to the **Register Descriptions** on page 5.

   e. Set the SPI clock phase (**CPHA**, byte 0, bit [2]) and polarity bits (**CPOL**, byte 0, bit [3]). Refer to the **Phase and Polarity** on page 11.

   f. Enable the SPI interrupts by setting the **SPIE** bit (byte 0, bit [7]) to "1".

   g. To enable the bidirectional mode of operation, set the **SPC0** bit (byte 1, bit [0]) to "1" and the **BIDIROE** bit (byte 1, bit [3]) to "1". For normal mode, set both the bits to "0".

2. If applicable, write to the optional preamble register (offset 0x08) with the desired Opcode/Address.

3. An external Slave must be selected by asserting the corresponding bit in the Slave Select Register (offset 0x14).

   a. Set the corresponding slave bit to "1" to select the external slave device. For example, to select the slave device connected to the SS_bar [7], set the **S7** bit (byte 0, bit [8]) to "1".

4. Write all the data bytes to be transmitted to the **WRITE FIFO PORT** (offset 0x18). For example, to transmit eight bytes of data, the WRITE FIFO PORT is written to twice.

5. Start data transfer by writing to the SPI Transfer Register (offset 0x0C).

   NOTE: Steps a through d can be performed with a single write to address offset 0x0C.

a. For a write operation, set the **W** bit (byte 0, bit [1]) to "1" and the **R** bit (byte 0, bit [2]) to "0".

b. Set the preamble register byte enable bits by setting the corresponding **BE** [3:0] bits to "1". For example, if the two most significant bytes of the preamble register are to be transmitted, set **BE** [3:2] (byte 2, bit [3:2]) to "11". If no preamble bytes are required, set **BE** [3:0] to "0000".

c. Configure the number of bytes to be transmitted by setting **data_bytes**[15:3] to the appropriate value. For example, to write 10 bytes, the value of **data_bytes**[15:3] should be "0000000001010".

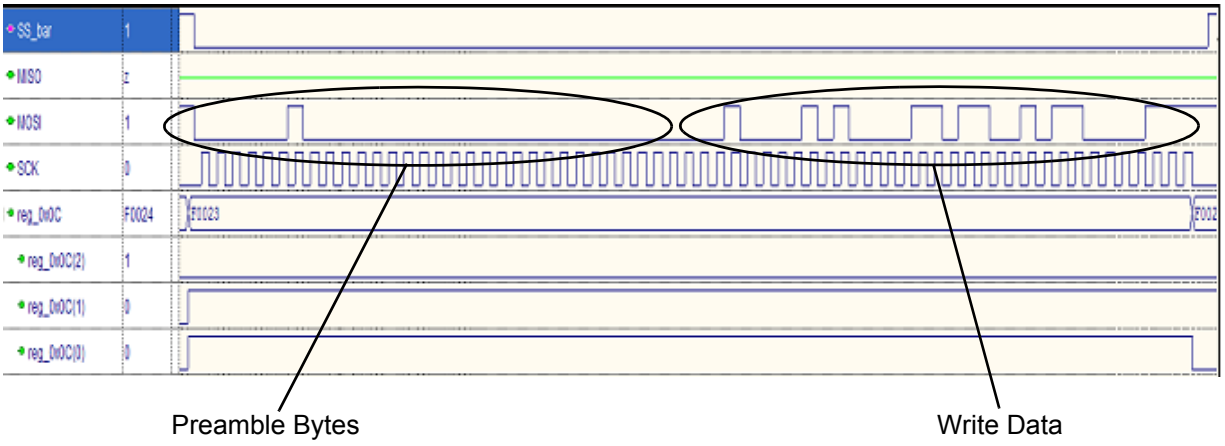
d. To start the write process, set the **S/Sp** bit (byte 0, bit [0]) to "1".

6. After the write transaction is completed, (i.e., the desired number of data bytes have been transmitted – 10 bytes for the above example), the SPI controller generates an interrupt signal to the processor, **IW** (offset 0x10, byte 0, bit 0).

   NOTE: This interrupt signal also automatically clears the S/Sp bit (byte 0, bit [0]) to "0".

   To acknowledge the interrupt, the processor must read the SPI interrupt register (offset 0x100) to clear the write interrupt flag.

Figure 5: Master Write Operation



Supported Operating Systems

The SPI Host Controller Proven System Block (PSB) supports the following operating systems:

- Windows® CE
- Windows Mobile®
- Linux®

Contact Information

Phone: (408) 990-4000 (US)
(905) 940-4149 (Canada)
+(44) 1932-57-9011 (Europe)
+(852) 2297-2297 (Asia)
E-mail: info@quicklogic.com
Sales: www.quicklogic.com/sales
Support: www.quicklogic.com/support
Internet: www.quicklogic.com

Revision History

Revision	Date	Originator and Comments
A	March 2008	Tina Durgia and Kathleen Murchek

Notice of Disclaimer

Copyright and Trademark Information